Kernel Logistic PLS: a new tool for complex classification

Arthur Tenenhaus[1,2], Alain Giron[1], Gilbert Saporta[3], and Bernard Fertil[1]

[1] INSERM U678, CHU Pitié-Salpêtrière, Paris, France
(e-mail: arthur.tenenhaus@imed.jussieu.fr,
alain.giron@imed.jussieu.fr,
bernard.fertil@imed.jussieu.fr)
[2] KXEN research, Suresnes, France
[3] CNAM – Conservatoire National des Arts et Métiers, France
(e-mail: saporta@cnam.fr)

Abstract. "Kernel Logistic PLS" (KL-PLS), a new tool for classification with performances similar to the most powerful statistical methods is described in this paper. KL-PLS is based on the principles of PLS generalized regression and learning via kernel. The successions of simple regressions, simple logistic regression and multiple logistic regressions on a small number of uncorrelated variables that are computed within KL-PLS algorithm are convenient for the management of very high dimensional data. The algorithm was applied to a variety of benchmark data sets for classification and in all cases, KL-PLS demonstrates its competitiveness with other state-of-art classification method. Furthermore, leaning on statistical tests related to the logistic regression, KL-PLS allows the systematic detection of data points close to "support vectors" of SVM and thus reduces the computational charges of the SVM training algorithm without significant loss of accuracy.

Keywords: Classification, Kernel, PLS Generalized Regression.

1 Introduction

Given a set of labeled experiments $\{(x_i, y_i)\}_{i=1,\ldots,n}$, $x_i \in \mathbb{R}^{p\times 1}$ and $y_i \in \{-1, 1\}$, we would like to build a prediction rule which, based on the observations, allows a prediction of the label y_{new} of a new point x_{new}. The following notation is used throughout this paper: each data point x_i (respectively each response y_i) represents the i^{th} row of the data matrix X (respectively the i^{th} row of the column vector Y). In order to handle the "generally" high dimensionality of the input space, we propose to exploit principles of the Partial Least Square regression (PLS) [Wold *et al.*, 1982, Tenenhaus, 1998]. PLS regression creates a set of orthogonal latent variables (PLS component) $t_1, t_2, \ldots, t_m$, linear combinations of the original variables but, contrary to principal component analysis (PCA), use the target Y for their determination. The PLS components t_h is obtained from the following constraints (Tucker criteria):

$$\max_{t_h} \operatorname{cov}^2(t_h, Y) \;=\; \max_{w_h} \operatorname{cov}^2(X_{h-1} w_h, Y)$$

$$\text{such that } \|w_h\| = 1 \text{ and } t_h \text{ is orthogonal to } t_1, \ldots, t_{h-1},$$

where $X_0 = X$ and X_{h-1} is the residual of the regression of X on $t_1, \ldots, t_{h-1}$.

A least square regression is then performed to relate Y to the PLS components.

But PLS was not originally designed as a tool for classification. Thus, based on the algorithmic structure of PLS regression, the PLS logistic regression was proposed for classification task [Tenenhaus, 2002, Bastien *et al.*, 2004].

PLS regression is designed to operate with input data that are high-dimensional and highly correlated (PLS is very popular in the chemometrics field), such a situation encountered by the use of kernel function [Schölkopf and Smola, 2002]. Based on kernel techniques, Rosipal and Trejo have proposed a nonlinear extension of PLS regression, the Kernel PLS regression (KPLS regression) [Rosipal and Trejo, 2001]. The approach was subsequently extended to the kernel orthonormalized PLS for classification problems [Rosipal *et al.*, 2003] using Barker and Rayens approach [Barker and Rayens, 2003].

In this paper we present a non linear extension via kernel of the PLS logistic regression: Kernel Logistic PLS (KL-PLS). Following Bennet and Embrechts who demonstrated interest of directly exploit kernel within the framework of PLS regression [Bennett and Embrechts, 2003] and noting the close connection between KPLS and PLS regression of Y on the kernel K, [Appendix 1], we propose an algorithm directly based on the factorization of the kernel matrix.

Furthermore, thanks to the statistical tests related to logistic regression, KL-PLS allows detecting points close to "support vectors" (points used by the Support Vector Machines (SVM) to compute the decision boundary). It is therefore possible to select a subset of the training set that is sufficient to derive the SVM decision boundary.

2 Kernel Logistic PLS (KL-PLS)

2.1 Algorithm

Principle of KL-PLS is to compute orthogonal latent variables in the space induced by the kernel matrix before performing logistic regression in the derived feature space. Therefore, KL-PLS is a 3-step algorithm:

1. **Computation of the kernel matrix**
   Let X be the matrix comprising the p explanatory variables x_k, $k = 1, \ldots, p$ and Y a binary variable (the target) observed on n samples. Let K be the kernel matrix associated to X. A usual kernel is given below:

$$\text{Gaussian kernel: } K(x_i, x_j) = \exp\left(-\tfrac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

The dimension of the kernel matrix is $n \times n$. Each cell k_{ij} is a measure of similarity between the individuals i and j.

2. **Computation of the KL-PLS components**

2.1 ***Computation of the first KL-PLS component*** t_1

Step 1: Compute the regression coefficient a_{1j} of k_j in the logistic regression of Y on k_j, $j = 1, \ldots, n$
Step 2: Normalize the column vector a_1 made by a_{1j}'s: $w_1 = a_1/\|a_1\|$
Step 3: Compute the first KL-PLS component as $t_1 = Kw_1$

2.2 ***Computation of the*** h^{th} ***KL-PLS component*** t_h

Let assume that in the previous steps, the KL-PLS components $t_1, \ldots, t_{h-1}$ have been yielded. This block is designed to get variables which, in addition to - and orthogonally to - $t_1, \ldots, t_{h-1}$, hold residual information on Y. The h^{th} KL-PLS component is subsequently computed from the residual of the regression of k_j, $j = 1, \ldots, n$ on $t_1, \ldots, t_{h-1}$.

Step 1: Compute the residual $e_{h1}, \ldots, e_{hn}$ from the multiple regression of k_j, $j = 1, \ldots, n$ on $t_1, \ldots, t_{h-1}$. Let K_{h-1} be the matrix comprising $\epsilon_{h1}, \ldots, \epsilon_{hn}$.
Step 2: Compute the coefficients a_{hj} of e_{hj} in the logistic regression of Y on $t_1, \ldots, t_{h-1}$ and e_{hj}.
Step 3: Normalize the column vector a_h made by a_{hj}'s: $w_h = a_h/\|a_h\|$.
Step 4: Compute the h^{th} PLS component: $t_h = K_{h-1}w_h$.
Step 5: Express the component t_h in terms of K as $t_h = Kw_h^*$.

3. ***Logistic regression of*** Y ***on the*** m ***retained KL-PLS components***

$$P(Y = 1|K = k) = \frac{e^{\alpha_0 + \sum_{h=1}^{m} \alpha_h t_h}}{1 + e^{\alpha_0 + \sum_{h=1}^{m} \alpha_h t_h}} .$$

2.2 Remarks

2.3 Selection of the number of useful KL-PLS components

Computation of the KL-PLS component t_h may be simplified by setting non-significant regression coefficients a_{hj} to 0. Only variables that are significantly related to Y contribute to the computation of t_h. The number m of KL-PLS components to be retained may be chosen by cross-validation or by observing that the component t_{m+1} is not significant because none of the coefficients $a_{(m+1)j}$ is significantly different from 0.

2.4 Expression of KL-PLS component in term of original variables

Expression of PLS components in terms of original variables is a fundamental step to analyze new data. Indeed, let $Ktest$ be the new dataset. The matrix product $Ttest = Ktest \times W^*$ allows to compute the values of the KL-PLS components for the new dataset.

2.4.1 Computation of w_h^*

a. The first KL-PLS component is already expressed in terms of original variables : $t_1 = Kw_1$ and $w_1^* = w_1$.

b. The second KL-PLS component is expressed in terms of the residuals in the regression of the original variables on t_1. From $K = t_1p_1' + K_1$ and $t_2 = K_1w_2$ we get:

$$t_2 = K_1w_2 = (K - t_1p_1') = (K - Kw_1p_1')w_2 = K\underbrace{(I - w_1p_1')w_2}_{w_2^*} = Kw_2^*.$$

c. In a similar way, it can be shown that t_h is expressed in terms of the original variables as:

$$\begin{aligned} t_h = K_{h-1}w_h &= \left(K - \sum_{i=1}^{h-1} t_ip_i\right) \cdot w_h = \left(K - \sum_{i=1}^{h-1} Kw_i^*p_i'\right) \cdot w_h \\ &= K\underbrace{\left(I - \sum_{i=1}^{h-1} w_i^*p_i'\right) \cdot w_h}_{w_h^*} = Kw_h^*. \end{aligned}$$

3 Kernel Logistic PLS and detection of support vectors

3.1 Preliminary considerations

SVM was designed to find the "optimal separating hyperplane" i.e. the hyperplane whose minimal distance to the training examples is maximum (fig. 1) [Vapnik, 1998]. The optimal hyperplane is defined by a vector β and a scalar β_0 through the equation:

$$\arg\max_{\beta,\beta_0} \min\left\{\|x - x_i\| : x \in \mathbb{R}^n,\ (x^t\beta + \beta_0) = 0,\ i = 1, \ldots, n\right\}.$$

Points which "support" hyperplanes H_1 and H_2 are the "support vectors". Only support vectors take part in the construction of the SVM decision boundary. We propose an approach which is able to detect points, called "ambiguous points" thereafter, close to support vectors. This procedure is achieved by removing a subset of training examples with minimal impact on the SVM decision boundary position.

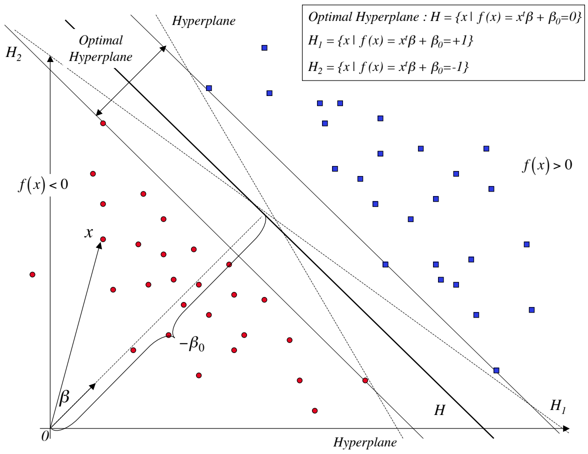
Optimal Hyperplane : H = {x | f (x) = x'β + β0=0}
H1 = {x | f (x) = x'β + β0=+1}
H2 = {x | f (x) = x'β + β0=-1}
Hyperplane
Optimal Hyperplane
H2
f(x) < 0
x
f(x) > 0
−β0
β
0
H
H1
Hyperplane

Fig. 1. Optimal separating hyperplane.

3.2 Detection of ambiguous points

During the construction of the first KL-PLS component, coefficients a_{1j} of k_j in the logistic regression of Y on each k_j, $j = 1, \ldots, n$ are computed. If a point j is, on the average, closer to the points belonging to its own group than to the points belonging to the other group, then k_j has, on the average, a larger value (in the case of Gaussian kernel) for the individuals belonging to the group containing j than for the other individuals. We can expect the regression coefficient a_{1j} to be highly significant in this situation. Consequently, it is proposed to label points associated to non-significant a_{1j} to the risk α (Wald test) as ambiguous.

The number of ambiguous points can, subsequently be controlled by increasing the risk α.

4 Results

4.1 Banana data projection onto the two first components found by KL-PLS

Banana data is a 2D dataset (two classes). 400 × 2 training set is associated to a 4,600 × 2 testing set. Figure 2 depicts projection of the original training and testing data onto the two first components found by KL-PLS (training data). A nice linear separation of the two classes can be seen in the feature space and logistic regression is adequate to achieve an efficient classification.

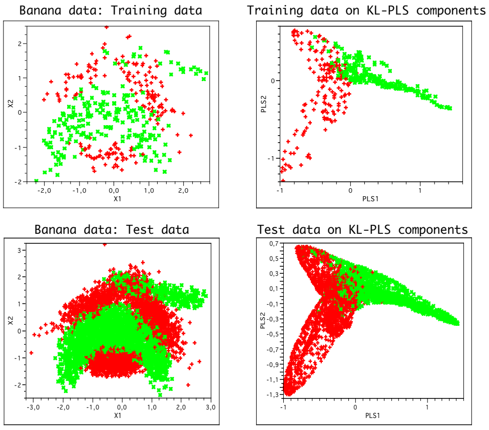

Fig. 2. Banana data depict onto the two first components found by kernel logistic PLS.

4.2 Benchmarks

The usefulness of KL-PLS was tested on several benchmark data sets (two-class classification) used in [Mika *et al.*, 1999] and [Rätsch *et al.*, 2001]. These datasets are available at http://ida.first.gmd.de/raetsch/data/benchmarks.htm. Each dataset consists of 100 different training and testing partitions. Several methods (KFD, SVM, KPLS-SVC) have already been used and results are presented in table 1. Baudat and Anouar have proposed a nonlinear extension of the Fisher Discriminant Analysis via "Kernel Trick": the Kernel Fisher Discriminant analysis (KFD) [Baudat and Anouar, 2000]. The kernel orthonormalized PLS + SVC (KPLS-SVC) is based on the kernel orthonormalized PLS method for dimensionality reduction followed by SVM on retained PLS components for classification [Rosipal *et al.*, 2003]. In all cases the Gaussian kernel was used. KL-PLS efficiency relies on the value of width of the Gaussian and the number of retained KL-PLS components Those values are selected based on the minimum classification error observed after five-fold cross validation on the first five training sets. Results of logistic regression (LR) are also presented. Results achieved for the 11 benchmarks demonstrate the efficiency of KL-PLS and its competitiveness with other state-of-the-art classification methods.

4.3 Ambigous points and support vectors

4.3.1 Simulated checkerboard A 4 × 4 checkerboard is represented in fig 3. Twenty-five uniformly points labeled according to checkerboard pattern

Table 1. Comparison of the mean and standard deviation classification errors (test set) for KFD [Mika *et al.*, 1999], SVM [Rätsch *et al.*, 2001], Kernel PLS-SVC [Rosipal *et al.*, 2003], Logistic Regression (LR) and KL-PLS. The last column provides the width of the Gaussian kernel and the number of retained KL-PLS components.

						KL-PLS
Data set	**KFD**	**SVM**	**KPLS-SVC**	**LR**	**KL-PLS**	**parameters**
Banana	10.8 ± 0.5	11.5 ± 0.5	10.5 ± 0.4	47.0 ± 4.48	10.7 ± 0.5	$(0.9, 10)$
B. Cancer	25.8 ± 4.6	26.0 ± 4.7	25.1 ± 4.5	27.5 ± 4.7	25.8 ± 4.4	$(50, 7)$
Diabetis	23.2 ± 1.6	23.5 ± 1.7	23.0 ± 1.7	23.3 ± 1.8	23.0 ± 1.7	$(60, 4)$
German	23.7 ± 2.2	23.6 ± 2.1	23.5 ± 1.6	24.0 ± 2.1	23.2 ± 2.1	$(20, 2)$
Heart	16.1 ± 3.4	16.0 ± 3.3	16.5 ± 3.6	16.9 ± 2.9	16.0 ± 3.2	$(20, 3)$
Ringnorm	1.49 ± 0.12	1.66 ± 0.12	1.43 ± 0.10	25.3 ± 0.8	1.44 ± 0.09	$(200, 2)$
F. Solar	33.2 ± 1.7	32.4 ± 1.8	32.4 ± 1.8	34.6 ± 3.7	32.7 ± 1.8	$(12, 1)$
Thyroid	4.20 ± 2.07	4.80 ± 2.19	4.39 ± 2.1	10.3 ± 2.7	4.35 ± 1.99	$(15, 6)$
Titanic	23.2 ± 2.06	22.4 ± 1.0	22.4 ± 1.1	22.7 ± 1.1	22.4 ± 0.04	$(300, 2)$
Twonorm	2.61 ± 0.15	2.96 ± 0.23	2.34 ± 0.11	3.81 ± 0.53	2.37 ± 0.10	$(40, 1)$
Waveform	9.86 ± 0.44	9.88 ± 0.43	9.58 ± 0.36	13.48 ± 0.7	9.74 ± 0.46	$(15, 4)$

was generated within each square. Fig. 3 depicts the projection of the 4×4 checkerboard from both classes onto the two first components found by KL-PLS. A nice separation of the two classes can be seen. Note that support vectors and ambiguous (blue circles) are pretty close.

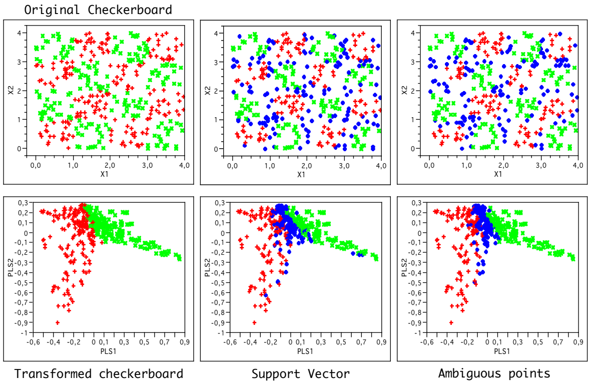

Fig. 3. Comparison between Support Vectors and Ambiguous Points (blue circles).

4.3.2 Selection of ambiguous points (banana data) The SVM decision boundary only depends on the support vectors. In order to evaluate

Table 2. Confusion matrix between Support Vectors and ambiguous points.

	Ambiguous points	**Non ambiguous points**	**Total**
Support vector	112	36	148
Non support vector	23	229	252
Total	135	265	400

proximity between ambiguous points and support vectors, SVM was trained on "α-selected" ambiguous points. Results were compared to those obtain by SVM (full training set).

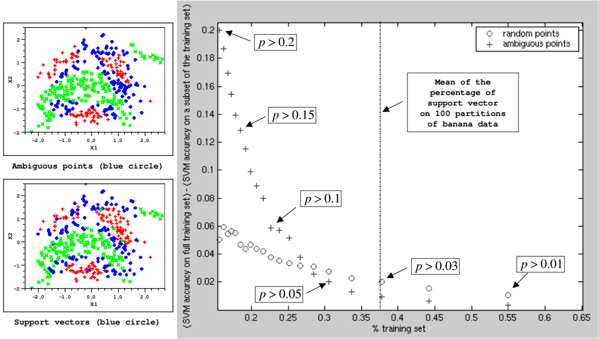

Fig. 4. Efficiency of SVM classification as function of the size of the training set. o - randomly selected points for the training set. + - points selected with respect to their significant level (p).

The following operations were carried out:

i. We compute the mean test set classification error based on SVM trained on full training set on the 100 partitions of banana data.
ii. For each $\alpha = \{0.01, 0.02, \ldots, 0.2\}$, KL-PLS was trained on the 100 partitions of banana data. It allows detection of ambiguous points for each partition. Then, SVM is trained on ambiguous points for each partition. We compute the mean test set classification error.
iii. For each $\alpha = \{0.01, 0.02, \ldots, 0.2\}$, SVM was trained on randomly selected points in the same proportion as the ambiguous points related to this value of α for each partition. We compute the mean test set classification error.

SVM train on ambiguous point gives performances similar to SVM train on full training set when the number of ambiguous points is close to the number of support vectors. Syed *et al.* have shown that the discarding of even a small proportion of the support vectors can lead to a severe reduction in generalization performance [Syed *et al.*, 1999]. They stated that this implies that the support vector set chosen by SVM is a minimal set; this can explain the behavior of the (blue - cross) curve (fig. 4) when considering low numbers of ambiguous points.

5 Discussion and conclusion

Performances of KL-PLS are equivalent to the most powerful classification methods such as SVM, KPLS-SVC or KFD. This algorithm is very simple to implement since it is solely composed of ordinary least square and logistic regressions. Furthermore, it is possible to compute KL-PLS components only by considering individual column vectors of the kernel matrix. These properties make possible to highlight 3 interests of KL-PLS:

a. KL-PLS does not require the full kernel matrix in memory but the columns of the kernel individually.
b. Inversions of small dimension matrices (number of KL-PLS components +1) take place in the algorithm.
c. The introduction of intercept when constructing the latent variables, avoid the kernel centering method proposed by Wu *et al.* [Wu *et al.*, 1997].

⇒ KL-PLS allows management of very high dimensional data.

Furthermore, direct factorization of the kernel matrix offers 2 advantages:

a. K does not need to be square
b. K does not need defining a dot product in the feature space induced by the "kernel trick". The Mercer's conditions (positive definite) are subsequently not required.

⇒ K just need to contain similarity measures.

Moreover, Kernel-PCA is often used as a preliminary step for dimensional reduction prior classification [Schölkopf *et al.*, 1998]. A more powerful goal-driven preprocessing is built in KL-PLS.

Lastly, leaning on Wald tests related to the logistic regression, it is possible to detect "ambiguous points" close to support vectors. This approach specifically selects examples from the training set close to support vectors. SVM computational charges are consequently reduced without jeopardizing classification.

Works in progress comprise the extension of KL-PLS approach to the multi-class classification problems, the study of the relationship between "ambiguous points" and "Support Vectors" and the extension of Kernel Logistic PLS to the kernel generalized PLS via generalized linear model.

Acknowledgment This work was supported by KXEN, a global analytic software company.

References

[Barker and Rayens, 2003]M. Barker and W. S Rayens. Partial least square for discrimination. *Journal of Chemometrics*, pages 166–173, 2003.

[Bastien *et al.*, 2004]P. Bastien, V. E. Vinzi, and M Tenenhaus. Pls generalized linear regression. *Computational Statistics & data analysis*, 2004.

[Baudat and Anouar, 2000]G. Baudat and F Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, pages 2385–2404, 2000.

[Bennett and Embrechts, 2003]K. P. Bennett and M. J Embrechts. An optimization perspective on kernel partial least squares regression, advances in learning theory: Methods, models and applications. *NATO Sciences Series III: Computer & Systems Sciences*, pages 227–250, 2003.

[Höskuldsson, 1988]A Höskuldsson. Pls regression methods. *Journal of Chemometrics*, pages 211–228, 1988.

[Mika *et al.*, 1999]S. Mika, G. Rätsch, J. Weston, Schölkopf B., and K. R Muller. Fischer discriminant analysis with kernels. *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[Rätsch *et al.*, 2001]G. Rätsch, T. Onoda, and K.R Muller. Soft margin for adaboost. *Machine Learning*, pages 287–320, 2001.

[Rosipal and Trejo, 2001]R. Rosipal and L.J Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2001.

[Rosipal *et al.*, 2003]R. Rosipal, L.J. Trejo, and B Matthews. Kernel pls-svc for linear and nonlinear classification. In *Proceeding of the twentieth international conference on machine learning (ICML-2003)*, 2003.

[Schölkopf and Smola, 2002]B. Schölkopf and A. J Smola. *Learning with kernel - Support Vector Machines Regularization, Optimization and Beyond.* The MIT Press, 2002.

[Schölkopf *et al.*, 1998]B. Schölkopf, A.J. Smola, and K.R Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[Syed *et al.*, 1999]N.A. Syed, H. Liu, and K. K Sung. Incremental learning with support vector machines. In *Proceeding of Workshop on Support Vector Machines at International Joint Conference on Artificial Intelligence*, 1999.

[Tenenhaus, 1998]M Tenenhaus. *La Régression PLS.* Éditions Technip, 1998.

[Tenenhaus, 2002]A Tenenhaus. La régression logistique pls validée par bootstrap. In *Mémoire de DEA de Statistique, Université Pierre et Marie Curie*, 2002.

[Vapnik, 1998]V Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[Wold *et al.*, 1982]S. Wold, L. Martens, and H Wold. The multivariate calibration problem in chemistry solved by the pls method. In *Conf. Matrix Pencils, Ruhe A. & Kåstrøm B, Lecture Notes in Mathematics*, pages 286–293. Springer Verlag, 1982.

[Wu *et al.*, 1997]W. Wu, D.L. Massart, and S de Jong. The kernel pca algorithms for wide data – part ii: Fast cross validation and application in classification of nir data. *Chemometrics and Intelligent Laboratory Systems*, pages 271–280, 1997.

6 Appendix: KPLS and PLS regression of Y on K

6.1 Kernel PLS (KPLS)

Höskuldsson shows that the weights vector w_1^{PLS} corresponds to the eigenvector associated to the greatest eigenvalue of the matrix $X'YY'X$ [Höskuldsson, 1988]. The first PLS component is then $t_1^{PLS} = Xw_1^{PLS}$.

$\Rightarrow X'YY'Xw_1^{PLS} = \lambda w_1^{PLS}$

$\Leftrightarrow XX'YY'\underbrace{Xw_1^{PLS}}_{t_1^{PLS}} = \lambda\underbrace{Xw_1^{PLS}}_{t_1^{PLS}}$

The first PLS component is the eigenvector associated to the greatest eigenvalue of $XX'YY'$.

Within the framework of PLS 1: $Y \in \mathbb{R}^n$.

$\Rightarrow Y'Xw_1^{PLS}$ is a scalar

$\Rightarrow t_1^{PLS}$ is proportional to $XX'Y$ and thus, we can rigorously be reduced to the framework of the kernel trick, giving arise to Kernel PLS; and write that $t_1^{KPLS} = KY$.

6.2 PLS regression of Y on K (DK-PLS)

In a similar way, the weight vector w_1^{DK-PLS} corresponds to the eigenvector associated to the greatest eigenvalue of the matrix $K'YY'K$. The first DK-PLS component is then $t_1^{DK-PLS} = Kw_1^{DK-PLS}$.

$\Rightarrow K'YY'Kw_1^{DK-PLS} = \lambda w_1^{DK-PLS}$

$\Leftrightarrow KK'YY'\underbrace{Kw_1^{DK-PLS}}_{t_1^{DK-PLS}} = \lambda\underbrace{Kw_1^{DK-PLS}}_{t_1^{DK-PLS}}.$

The first DK-PLS component is the eigenvector associated to the greatest eigenvalue of $KK'YY'$.

Within the framework of PLS 1: $Y \in \mathbb{R}^n$.

$\Rightarrow Y'Kw_1^{DK-PLS}$ is a scalar

$\Rightarrow t_1^{DK-PLS}$ is proportional to $KK'Y$ being, by construction, symmetric

$\Rightarrow t_1^{DK-PLS}$ is proportional to K^2Y.

In a similar way, $t_h^{K-PLS} = K_{h-1}Y$ and $t_h^{DK-PLS} = K_{h-1}^2 Y = K_{h-1}w_h^{DK-PLS}$ where $w_h^{DK-PLS} = K_{h-1}Y$ and K_{h-1} is the matrix comprising the p residual vector $e_{h1}, \ldots, e_{hp}$ of the ordinary least square of $k_j,\ j = 1, \ldots, p$ on $t_1, \ldots, t_{h-1}$. Let us notice that $t_h^{K-PLS} = w_h^{DK-PLS}$.